AMENDMENTS TO THE CLAIMS

1. (Original) A method comprising:

setting a flag for a processor via a first instruction, the first instruction being either a direct update instruction or an indirect update instruction;

if the setting of the flag is by a direct update instruction, executing a succeeding second instruction that reads the flag prior to completion of the first instruction; and

if the setting of the flag is by an indirect update instruction, delaying the second instruction until after completion of the first instruction.

2. (Original) The method of claim 1, wherein setting the flag by a direct update instruction comprises storing a value for the flag in a renamer.

3. (Original) The method of claim 2, wherein setting the flag by an indirect update instruction comprises storing a value in a buffer and storing the value in the renamer upon retirement of the indirect update instruction.

4. (Original) The method of claim 3, further comprising checking the value of a register scoreboard prior to accessing the flag.

5. (Original) The method of claim 4, wherein executing an indirect update instruction comprises setting the register scoreboard prior to storing the value in the renamer and clearing the register scoreboard after storing the value in the renamer.

6. (Original) The method of claim 1, further comprising storing the value for the flag in shadow logic.

7. (Original) The method of claim 6, wherein the shadow logic is enabled if the value was provided by a direct update instruction.

8. (Currently Amended) The method of claim 7, wherein the shadow logic is disabled if the value was not provided by a direct update instruction.

9. (Currently Amended) A processor comprising:

an execution unit to execute instructions; and

a renamer, the renamer to rename a flag register and store the value for the flag register;

the value of the flag register being set by one of a plurality of processes, the processes including directly setting the flag register by a first instruction or setting the flag register to ~~an~~ a data value obtained by a second instruction; and

a succeeding third instruction being executed without being stalled if the value of the flag register was set by the first instruction and being stalled until conclusion of the second instruction if the value of the flag register is set by the second instruction.

10. (Original) The processor of claim 9, further comprising a scoreboard register, the first instruction and the second instruction checking the scoreboard register before setting the flag register.

11. (Original) The processor of claim 10, wherein the first instruction and the second instruction delay storage of the flag register if the scoreboard register is enabled.

12. (Original) The processor of claim 11, wherein execution of the second instruction includes setting the scoreboard register before setting the flag register and clearing the scoreboard register after setting the flag register.

13. (Original) The processor of claim 9, further comprising shadow logic to store values for the flag register.

14. (Original) The processor of claim 13, wherein the shadow logic includes a validity register to indicate validity of stored values for the flag register.

15. (Original) The processor of claim 14, wherein the validity register is enabled if a value for the flag register is provided by the first instruction and the validity register is disabled if the value for the flag register is provided by the second instruction.

16. (Original) The processor of claim 9, wherein the flag register is one of an interrupt flag or a direction flag.

17. (Original) The processor of claim 16, wherein the first instruction is an instruction to set or clear the flag register.

18. (Original) The processor of claim 9, wherein the second instruction is an instruction to pop a data value from a memory stack.

19. (Currently Amended) A system comprising:

a bus;

a flash memory coupled to the bus; and

a processor coupled to the bus, the processor comprising:

an execution unit to execute instructions; and

a renamer, the renamer to rename a flag register and store the value for the flag register;

the value of the flag register being set by one of a plurality of processes, the processes including directly setting the flag register by a first instruction or setting the flag register to ~~an~~ a data value obtained by a second instruction; and

a succeeding third instruction being executed without being stalled if the value of the flag register was set by the first instruction and being stalled until conclusion of the second instruction if the value of the flag register is set by the second instruction.

20. (Original) The system of claim 19, wherein the processor further comprises a scoreboard register, the first instruction and the second instruction checking the scoreboard register before setting the flag register.

21. (Original) The system of claim 20, wherein the first instruction and the second instruction delay storage of the flag register if the scoreboard register is enabled.

22. (Original) The system of claim 21, wherein execution of the second instruction includes setting the scoreboard register before setting the flag register and clearing the scoreboard register after setting the flag register.

23. (Original) The system of claim 19, wherein the processor further comprises shadow logic to store values for the flag register.

24. (Original) The system of claim 23, wherein the shadow logic includes a validity register to indicate validity of stored values for the flag register.

25. (Original) The system of claim 24, wherein the validity register is enabled if a value for the flag register is provided by the first instruction and the validity register is disabled if the value for the flag register is provided by the second instruction.

26. (Original) The system of claim 19, wherein the flag register is one of an interrupt flag or a direction flag.

27. (Original) The system of claim 26, wherein the first instruction is an instruction to set or clear the flag register.

28. (Original) The system of claim 19, wherein the second instruction is an instruction to pop a data value from a memory stack.